Package 'rvmethod'

July 27, 2020

Type Package

Title Radial Velocity Method for Detecting Exoplanets

Version 0.1.1

Author Parker Holzer

Maintainer Parker Holzer <parker.holzer@yale.edu>

Description Has various functions designed to implement the Hermite-Gaussian Radial Velocity (HGRV) estimation approach of Holzer et al. (2020) <arXiv:2005.14083>, which is a particular application of the radial velocity method for detecting exoplanets. The overall approach consists of four sequential steps, each of which has a function in this package: (1) estimate the template spectrum with the function estimate_template(), (2) find absorption features in the estimated template with the function findabsorptionfeatures(), (3) fit Gaussians to the absorption features with the function Gaussfit(), (4) apply the HGRV with simple linear regression by calling the function hgrv(). This package is meant to be open source. But please cite the paper Holzer et al. (2020) <arXiv:2005.14083> when publishing results that use this package.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

Imports parallel, locfit, assertthat

Depends R (>= 2.10)

NeedsCompilation no

Repository CRAN

Date/Publication 2020-07-27 15:40:02 UTC

R topics documented:

`estimate_template` *Estimate the Template Spectrum*

Description

This function uses local quadratic regression to estimate the template spectrum from a collection of observed spectra from a star as described in Holzer et al. (2020). All observed spectra are assumed to be normalized. The bandwidth is chosen locally through generalized cross-validation. We **strongly** recommend using parallel computing for this function. Therefore, the `cores` argument has the default value of 19.

Usage

```
estimate_template(
  SPECTRA,
  min_wvl = NULL,
  max_wvl = NULL,
  bandwidth_bnds = c(0.017, 0.05),
  min_count = 100,
  cores = 19
)
```

Arguments

`SPECTRA`	a list of all observed spectra to use in estimating the template. Each observed spectrum should have the format of being a list with the following names (or a dataframe with the following columns): “Wavelength" and “Flux".
`min_wvl`	a number that indicates the minimum wavelength for the estimated template
`max_wvl`	a number that indicates the maximum wavelength for the estimated template
`bandwidth_bnds`	a vector of length 2 that gives the interval of bandwidth values (in the same units as the wavelength of the spectra) to be considered in the generalized cross-validation

min_count	the minimum number of data points required for local regression to be done on a given wavelength chunk
cores	the number of cores to parallelize over (if set to 1, no parallelizing is done)

Value

a list with the following elements:

Wavelength	the wavelength axis of the estimated template
Flux	the normalized flux of the estimated template
Chunk_bounds	a list of length 2 vectors that give the wavelength bounds for each chunk for which the smoothing was done on
Bandwidths	the bandwidths chosen for each of the chunks
Std_err	the standard errors of the estimated normalized flux that can be used for prediction confidence intervals

Examples

```
data(spectra)
plot(spectra[[1]]$Wavelength, spectra[[1]]$Flux, col='gray', type='l')
for(spec in spectra){
 lines(spec$Wavelength, spec$Flux, col='gray')
}
tempest = estimate_template(spectra, cores = 1)
lines(tempest$Wavelength, tempest$Flux, col='red')
```

---

findabsorptionfeatures

Find Absorption Features in a Spectrum

Description

This function applies the Absorption Feature Finder algorithm (Algorithm 1 in Holzer et. al 2020) to find absorption features in a high signal-to-noise, normalized, spectrum. For a spectrum that covers more than 100 Angstroms, it is recommended to parallelize it by setting the cores argument to be greater than 1.

Usage

```
findabsorptionfeatures(
  wvl,
  flux,
  pix_range = 7,
  gamma = 0.01,
  alpha = 0.05,
  minlinedepth = 0,
  cores = 1
)
```

Arguments

wvl	vector of wavelengths in the spectrum
flux	vector of normalized flux in the spectrum (must have the same length as wvl)
pix_range	integer that specifies the window size in units of pixels to use in the moving linear regression
gamma	significance level used in finding local minima
alpha	significance level used in estimating wavelength bounds of features (**Note:** this must be larger than gamma)
minlinedepth	minimum depth required for found absorption features to be returned
cores	number of cores to parallelize over (if set to 1, no parallelizing is done)

Value

a list with the following components:

wvbounds	a list of length 2 vectors that each give the lower and upper bounds of found absorption features
min_wvl	a vector of the wavelengths at which the minimum flux is achieved for each found absorption feature
min_flx	a vector of the minimum flux for each found absorption feature
max_flx	a vector of the maximum flux for each found absorption feature

Examples

```
data(template)
ftrs = findabsorptionfeatures(template$Wavelength,
                              template$Flux,
                              pix_range = 8, gamma = 0.05,
                              alpha = 0.07, minlinedepth = 0.015)
plot(template$Wavelength, template$Flux,
     type='l', xlab = "Wavelength", ylab = "Flux")
for(i in 1:length(ftrs$wvbounds)){
  lines(ftrs$wvbounds[[i]],
        c(1,1) - 0.01*rep(i%%2,2), col=3)
}
```

---

fit3gauss *Fit Gaussians to Three Absorption Features*

Description

This function takes a spectrum and the wavelength bounds of three neighboring absorption features and uses the functions gauss1func, gauss2func, and/or gauss3func to fit Gaussians to them simultaneously. The final fit is the first of the following five outcomes for which the nonlinear regression algorithm converges: (i) all three Gaussians, (ii) the left two Gaussians, (iii) the right two Gaussians, (iv) just the middle Gaussian, (v) the middle Gaussian with an amplitude of 0. Only the fit parameters for the middle Gaussian are returned.

Usage

```
fit3gauss(wvl, flx, bnds0, bnds1, bnds2)
```

Arguments

wvl	the vector of wavelengths of the spectrum to fit to
flx	the vector of normalized flux of the spectrum to fit to
bnds0	a vector of length 2 with the lower and upper bounds of the left absorption feature
bnds1	a vector of length 2 with the lower and upper bounds of the middle absorption feature
bnds2	a vector of length 2 with the lower and upper bounds of the right absorption feature

Value

a list with three components:

mu	the fitted value of the center parameter for the middle Gaussian
sig	the fitted value of the spread parameter for the middle Gaussian
amp	the fitted value of the amplitude parameter for the middle Gaussian

Examples

```
x = seq(5000, 5003, length.out=200)
y = gauss3func(x, 0.3, 0.5, 0.4, 5001.5, 5002, 5000.4, 0.1, 0.1, 0.13)
y = rnorm(200, mean=y, sd=0.01)
plot(x, y)
abline(v=c(5000.8, 5001.2, 5001.75, 5002.3))
pars = fit3gauss(x, y, c(5000, 5000.8), c(5001.2, 5001.75), c(5001.75, 5002.3))
fitted = gauss1func(x, pars$amp, pars$mu, pars$sig)
lines(x, fitted, col=2)
```

---

gauss1func *A Single Gaussian Absorption Feature*

Description

This function returns a Gaussian absorption feature with continuum 1.0 and a specified amplitude, center, and spread.

Usage

```
gauss1func(x, a1, mu1, sig1)
```

Arguments

x	the vector of values at which to evaluate
a1	the amplitude of the feature
mu1	the center of the feature
sig1	the spread of the feature (must be greater than 0)

Value

vector of values of the specified inverted Gaussian

Examples

```
x = seq(5000, 5003, length.out=200)
y = gauss1func(x, 0.3, 5001.5, 0.1)
plot(x, y)
```

---

gauss2func *Two Gaussian Absorption Features*

Description

This function returns two Gaussian absorption features, both with continuum 1.0 and each with a specified amplitude, center, and spread.

Usage

```
gauss2func(x, a1, a2, mu1, mu2, sig1, sig2)
```

Arguments

x	the vector of values at which to evaluate
a1	the amplitude of the first feature
a2	the amplitude of the second feature
mu1	the center of the first feature
mu2	the center of the second feature
sig1	the spread of the first feature (must be greater than 0)
sig2	the spread of the second feature (must be greater than 0)

Value

vector of values of the two specified inverted Gaussians

Examples

```
x = seq(5000, 5003, length.out=200)
y = gauss2func(x, 0.3, 0.5, 5001.5, 5002, 0.1, 0.1)
plot(x, y)
```

---

gauss3func *Three Gaussian Absorption Features*

Description

This function returns three Gaussian absorption features, both with continuum 1.0 and each with a specified amplitude, center, and spread.

Usage

```
gauss3func(x, a1, a2, a3, mu1, mu2, mu3, sig1, sig2, sig3)
```

Arguments

x	the vector of values at which to evaluate
a1	the amplitude of the first feature
a2	the amplitude of the second feature
a3	the amplitude of the third feature
mu1	the center of the first feature
mu2	the center of the second feature
mu3	the center of the third feature
sig1	the spread of the first feature (must be greater than 0)
sig2	the spread of the second feature (must be greater than 0)
sig3	the spread of the third feature (must be greater than 0)

Value

vector of values of the three specified inverted Gaussians

Examples

```
x = seq(5000, 5003, length.out=200)
y = gauss3func(x, 0.3, 0.5, 0.4, 5001.5, 5002, 5000.4, 0.1, 0.1, 0.13)
plot(x, y)
```

`Gaussfit` *Fit Gaussians to Absorption Features of a Normalized Spectrum*

Description

This function takes a spectrum, which ideally is a high signal-to-noise template spectrum estimated with the `estimate_template` function, and the absorption features found with the `findabsorptionfeatures` function and uses nonlinear least-squares to fit Gaussians to the features. This follows the procedure described in Holzer et al. (2020).

Usage

```
Gaussfit(wvl, flx, ftrs, cores = 1, mse_max1 = 0.00014, mse_max2 = 1e-04)
```

Arguments

`wvl`	vector of wavelengths of the spectrum
`flx`	vector of normalized flux of the spectrum
`ftrs`	a list of length 2 vectors that each give the lower and upper bounds of found absorption features. The `wvbounds` component of the `findabsorptionfeatures` function output is designed to be this.
`cores`	the integer count of cores to parallelize over. If set to 1, no parallelization is done.
`mse_max1`	the maximum mean squared error required for a fit from one Gaussian to be considered a good fit for an absorption feature
`mse_max2`	the maximum mean squared error required for a fit of two Gaussians to be considered a good fit for an absorption feature

Value

a list with the following components:

`parameters`	a dataframe with the wavelength bounds, fitted amplitude, fitted center, fitted spread, and fit type for absorption features with a good fit. A fit type of 0 indicates that the feature had a good fit of a single Gaussian. A fit type of 1 indicates that the feature did not have a good fit with a single Gaussian initially, but after fitting with two it did.
`fitted`	the vector of fitted values (with the same length as the `wvl` parameter) using only the features that produced a good fit.
`goodfits`	a vector of the indices for which rows in the `ftrs` parameter were well fitted with either 1 or 2 Gaussians at the end
`mse`	a vector with the mean squared error for each of the features in the `ftrs` parameter using the final fitted values

Examples

```
data(template)
ftrs = findabsorptionfeatures(template$Wavelength,
                              template$Flux,
                              pix_range = 8, gamma = 0.05,
                              alpha = 0.07, minlinedepth = 0.015)
gapp = Gaussfit(template$Wavelength, template$Flux, ftrs)
plot(template$Wavelength, template$Flux)
lines(template$Wavelength, gapp$fitted, col=2)
```

---

gaussfunc *Gaussian Function*

Description

This function returns the unnormalized (height of 1.0) Gaussian curve with a given center and spread.

Usage

```
gaussfunc(x, mu, sigma)
```

Arguments

x	the vector of values at which to evaluate the Gaussian
mu	the center of the Gaussian
sigma	the spread of the Gaussian (must be greater than 0)

Value

vector of values of the Gaussian

Examples

```
x = seq(-4, 4, length.out = 100)
y = gaussfunc(x, 0, 1)
plot(x, y)
```

HG1 *Evaluate the First-Degree Generalized Hermite-Gaussian Function*

Description

This function evaluates the first-degree Hermite-Gaussian function with a general center and spread.

Usage

```
HG1(x, mu, sig)
```

Arguments

x	the vector of values at which to evaluate the function
mu	the center parameter of the function
sig	the spread parameter of the function

Value

vector of values of the specified first-degree generalized Hermite-Gaussian function

Examples

```
x = seq(50, 60, length.out=100)
y = HG1(x, 55, 1)
plot(x, y)
```

hgrv *Apply the Hermite-Gaussian Radial Velocity (HGRV) Estimation Method*

Description

This function applies the HGRV method as given in Holzer et al. (2020) to a given observed spectrum, using the estimated template from the `estimate_template` function and the `parameters` component of the output from the `Gaussfit` function. The result is an estimate of the relative radial velocity present in the observed spectrum in units of m/s.

Usage

```
hgrv(obs_wvl, obs_flx, tmp_wvl, tmp_flx, Features, obs_err = NULL, cntm = NULL)
```

Arguments

obs_wvl	the vector of wavelengths of the observed spectrum
obs_flx	the vector of normalized flux of the observed spectrum
tmp_wvl	the vector of wavelengths of the template spectrum
tmp_flx	the vector of normalized flux of the template spectrum
Features	a dataframe with the wavelength bounds and fitted Gaussian parameters for each absorption feature. The parameters component of the output from the Gaussfit function provides this.
obs_err	the vector of uncertainties in the normalized flux of the observed spectrum (must be the same length as obs_wvl and obs_flx)
cntm	the vector of continuum values used to normalize the flux of the observed spectrum (must be the same length as obs_wvl and obs_flx)

Value

a list with the following components

rv	the estimated radial velocity in units of m/s
rv_err	the standard error of the estimated radial velocity in units of m/s
n	the number of data points used in the weighted linear regression
data	a list with the observed wavelengths (wvl), the difference flux (diff_flux), the explanatory variable constructed as a sum of first-degree generalized Hermite-Gaussian functions (hgvar), and the weights (weights) used in the regression.

Examples

```
data(template)
ftrs = findabsorptionfeatures(template$Wavelength,
                              template$Flux,
                              pix_range = 8, gamma = 0.05,
                              alpha = 0.07, minlinedepth = 0.015)
gapp = Gaussfit(template$Wavelength, template$Flux, ftrs)
data(observed_spec)
hgrv_output = hgrv(observed_spec$Wavelength, observed_spec$Flux,
                   template$Wavelength, template$Flux, gapp$parameters,
                   obs_err = observed_spec$Uncertainty)
plot(hgrv_output$data$hgvar, hgrv_output$data$diff_flux)
abline(a=0, b=hgrv_output$rv)
abline(a=0, b=hgrv_output$rv - 3*hgrv_output$rv_err, lty=2)
abline(a=0, b=hgrv_output$rv + 3*hgrv_output$rv_err, lty=2)
```

observed_spec *Observed spectrum for the star 51 Pegasi (HD 217014)*

Description

A small portion of one observed spectrum collected by EXPRES **Petersburg et. al (2020)**.

Usage

```
observed_spec
```

Format

A dataframe with 628 rows and the following 3 columns:

Wavelength the wavelength of the spectrum, in Angstroms

Flux normalized flux of the spectrum, unitless

Uncertainty the uncertainty of the flux measurements, unitless ...

Source

https://arxiv.org/abs/2003.08851

spectra *Observed spectra for the star 51 Pegasi (HD 217014)*

Description

56 observed spectra as collected by EXPRES **Petersburg et. al (2020)**. Only the subset of the spectrum between 5000 and 5005 Angstroms is given here.

Usage

```
spectra
```

Format

A list with 56 elements, each of which has 2 variables:

Wavelength the wavelength of the spectrum, in Angstroms

Flux normalized flux of the spectrum, unitless ...

Source

https://arxiv.org/abs/2003.08851

template *Estimated template spectrum for the star 51 Pegasi (HD 217014)*

Description

A small portion of the estimated template produced with the method of Holzer et. al (2020) on a set of 55 observed spectra from EXPRES.

Usage

```
template
```

Format

A data frame with 1893 rows and 2 variables:

Wavelength the wavelength of the spectrum, in Angstroms

Flux normalized flux of the spectrum, unitless ...

Source

https://arxiv.org/abs/2003.08851

wave_match *Adjust the wavelength solution of a spectrum*

Description

This function takes the wavelength and flux vectors of a normalized spectrum and uses cubic-spline interpolation to adjust the flux vector to match a new wavelength solution.

Usage

```
wave_match(wvl1, flx1, targetwvl)
```

Arguments

wvl1	vector of wavelengths for the spectrum to be interpolated
flx1	vector of normalized flux for the spectrum to be interpolated
targetwvl	vector of wavelengths to interpolate to.

Value

A vector of normalized flux for the spectrum at the targetwvl wavelengths. Only flux for targetwvl wavelengths that are contained by the wvl1 wavelengths are returned.

Examples

```
x = seq(0,10)
y = 5*sin(x + 2)
newx = seq(0.5, 9.5)
newy = wave_match(x, y, newx)
plot(x, y)
points(newx, newy, col=2, pch=19)
```

Index